Underriner HW2

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(haven)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(OOmisc)
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
data <- read_dta(file = "conf06.dta")
```

```
conf06 <- subset(data, data$nominee!="ALITO")
vars <- c("vote", "nominee", "sameprty", "qual", "lackqual", "EuclDist2", "strngprs") # vector of vars
conf <- conf06[vars] # retain only key vars from above object conf$numvote <- as.numeric(conf$vote)-1 #
```

1. Define an 80/20 train/test split.
2. Build logit classifier. Present the results in a useful way (e.g., a confusion matrix, etc.). Discuss the output in substantive terms.

```
test_vote = test$vote

logit <- glm(vote ~ EuclDist2 + qual  + strngprs + sameprty,
             data = train,
             family = binomial)

#dropped lack of qual cus obvoious its just the inverted figure of qual, or #multicollinear

logit.probs <- predict(logit,
                       newdata = test,
```

```
                      type="response")

logit.pred <- ifelse(logit.probs > 0.5, 1, 0)

table(logit.pred, test_vote)
```

```
##           test_vote
## logit.pred   0   1
##          0  36  16
##          1  50 660
```

```
mean(logit.pred == test_vote)
```

```
## [1] 0.9133858
```

As you can see above, this classifier has a good accuracy rating, correctly predicting the voting output of the test set roughly 90 percent of the time. Looking at the confusion matrix we can see that we correct label a yes vote (1) 653 times, and correctly label a no vote 38 times.

We had a false positve rate of 20/(20+38) of 34 percent, which is high (the high number of yes answers in our sample relative to no answers likey adds to this high rate).

3. Build an LDA classifier. Present the results in a useful way (e.g., a confusion matrix, etc.). Discuss the output in substantive terms.

```
## Linear Discriminant Analysis
lda <- lda(vote ~ EuclDist2 + qual  + strngprs + sameprty,
             data = train)

test_vote = test$vote

lda.pred <- predict(lda, newdata=test)

#data.frame(lda.pred)

# here is the confusion matrix
table(lda.pred$class, test_vote)
```

```
##    test_vote
##       0   1
##   0  41  23
##   1  45 653
```

```
# classification rate
mean(lda.pred$class == test_vote)
```

```
## [1] 0.9107612
```

```
lda
```

```
## Call:
## lda(vote ~ EuclDist2 + qual + strngprs + sameprty, data = train)
##
## Prior probabilities of groups:
##         0         1
## 0.1309485 0.8690515
##
## Group means:
```

```
##   EuclDist2      qual  strngprs  sameprty
## 0 0.3923202 0.5689975 0.3132832 0.1629073
## 1 0.1526028 0.8111310 0.6265106 0.6110272
##
## Coefficients of linear discriminants:
##                  LD1
## EuclDist2 -3.0752021
## qual       2.6186566
## strngprs   0.6082877
## sameprty   0.6868501
```

As you can see above, this classifier has a decent accuracy rating (nearly the same as the logit model), correctly predicting the voting output of the test set roughly 90 percent of the time. Looking at the confusion matrix we can see that we correct label a yes vote (1) 650 times, and correctly label a no vote 42 times.

We had a false positve rate of 23/(23+42) of 35 percent, which is high, one percentage point higher than the logit model. Again, the performance of LDA and logit here is similar.

Our false negative rate 47/(47+650) is 6.7 percent.

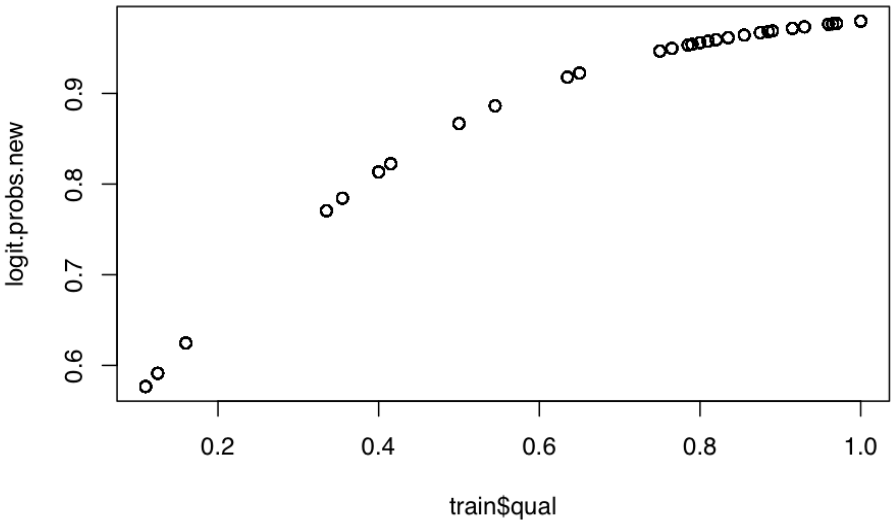
4.Calculate and plot the predicted probabilities from only the logit model, over the range of the nominees' perceived qualifications, holding all other variables at their mean values (hint: we did this in class). Describe these results in substantive terms (e.g., how does the probability of a yes vote change as the nominees' qualifications lessen?).

```
logit <- glm(vote ~ EuclDist2 + qual + strngprs + sameprty,
             data = train,
             family = binomial)

train.mean = with(train, data.frame(EuclDist2 = mean(EuclDist2),
                                    strngprs = mean(strngprs),
                                    sameprty = mean(sameprty),
                                    qual = qual))


logit.probs.new <- predict(logit,
                       newdata = train.mean,
                       type="response")

plot(train$qual, logit.probs.new)
```



As the perseived qualifcations of a nominee goes up there is a postive (with decaying quadratic growth) relationship between this perseption and a senator voting yes on their nomination.

5. Offer a couple paragraphs explaining all of your findings for the reader. For example, talk about different variables' impacts on the ability to correctly classify a Supreme Court nominee being supported. Or consider talking about how politicized the nominations process has become, and how we do (or do not) see this in these data. Just present a nice summary; short, to the point.

Here we are using data on Senators' voting patterns on Supreme Court nominees to gain insight into what causes them to confirm candidates. We are using four data points EuclDist2, a measure of the spatial ideological difference between the nominee and the senator, qual, the perceived qualifications of the nominee (we had a datapoint, lack qual, which was dropped as it would be perfectly multicollinear with qualification), and then two dummy variables, strngprs (which indicates the strength of the president, who is the one who puts forth supreme court nominations), and sameprty, which indicates if the nominee is of the same party as the senator. (there is certainly some multicollinear with this and EuclDist2, but there is enough intra party difference to still get useful explanatory power here).

From the magnitude of coefficients in the logit model its clear that EuclDist2 and qual are the two most important indicators (with the absolute value of the magnitude of the former slightly higher than the latter). This makes sense given that the closer to a senators politics a nominee is (as well as how well qualified they are for the job), the more likely they are to vote for their confirmation. While the graph is question 4 shows us that the prediction of a yes vote seems to highly clustered past the .75 qualification score, there are cases where the perceived quality can be quite low and there is still a $>.5$ chance of them being nominated, indicating that political alignment can overpower the worry about having someone who is a mediocre judge on the bench.

Given the nature of the nomination process, where a president is only going to put forth a nominee that they reasonably assume will get confirmed, our data set has a lot more yes votes then no votes. Because the model has much more yes votes to train on we have a much lower false negative rate than false positive rate, or put another way, we are much more likely to predict a yes vote where there isn't one than we are to predict a no vote when its actually a yes.

PART 2

```
library(wnominate)
```

```
##
## ## W-NOMINATE Ideal Point Package

## ## Copyright 2006 -2019

## ## Keith Poole, Jeffrey Lewis, James Lo, and Royce Carroll

## ## Support provided by the U.S. National Science Foundation

## ## NSF Grant SES-0611974
```

```
library(pscl)
```

```
house113 <- readKH(
"hou113kh.ord", # locate the .ord file saved locally dtl=NULL,
yea=c(1,2,3),
nay=c(4,5,6),
missing=c(7,8,9),
notInLegis=0,
desc="113th_House_Roll_Call_Data",
debug=FALSE
)
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 113th_House_Roll_Call_Data
## 445 legislators and 1202 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9
##  14576 295753 202943    290  21328
```

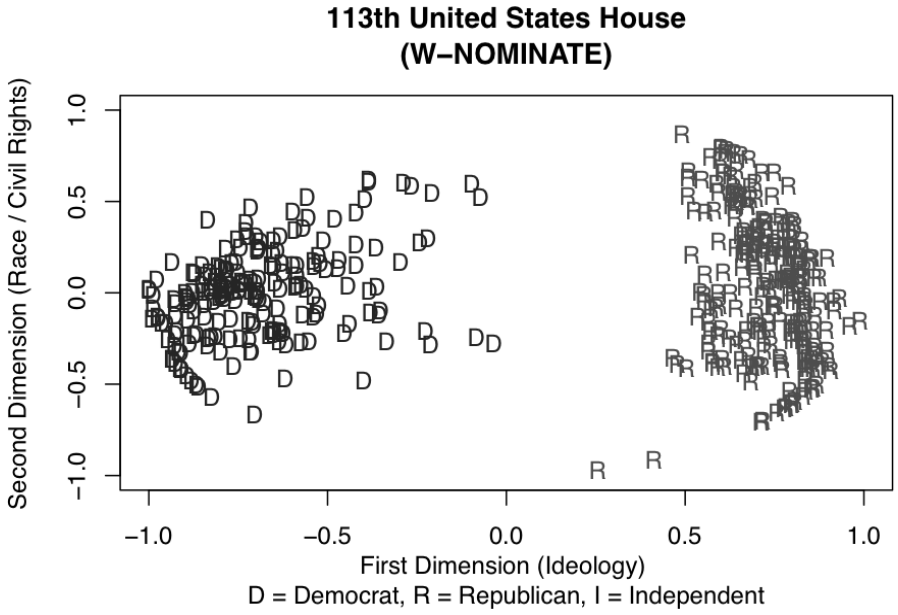
1. Fit a W-NOMINATE algorithm. Present a plot of members and discuss the results in substantive terms. What do you see?

```
wnom_result <- wnominate(house113,
                         dims = 2,
                         minvotes = 20,
                         lop = 0.025,
                         polarity = c(2,2))
```

```
##
## Preparing to run W-NOMINATE...
##
##  Checking data...
##
##      ... 1 of 445 total members dropped.
##
##      Votes dropped:
##      ... 181 of 1202 total votes dropped.
##
##  Running W-NOMINATE...
##
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Starting estimation of Beta...
##      Getting bill parameters...
##      Getting legislator coordinates...
```

```
##      Starting estimation of Beta...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Estimating weights...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Estimating weights...
##      Getting bill parameters...
##      Getting legislator coordinates...
##
##
## W-NOMINATE estimation completed successfully.
## W-NOMINATE took 185.571 seconds to execute.
```

```r
wnom1 <- wnom_result$legislators$coord1D
wnom2 <- wnom_result$legislators$coord2D
party <- house113$legis.data$party
plot(wnom1, wnom2,
     main="113th United States House\n(W-NOMINATE)",
     xlab="First Dimension (Ideology) \nD = Democrat, R = Republican, I = Independent",
     ylab="Second Dimension (Race / Civil Rights)",
     xlim=c(-1,1), ylim=c(-1,1), type="n")
points(wnom1[party=="D"], wnom2[party=="D"], pch="D", col="grey15")
points(wnom1[party=="R"], wnom2[party=="R"], pch="R", col="grey30")
points(wnom1[party=="Indep"], wnom2[party=="Indep"], pch="I", col="red")
```

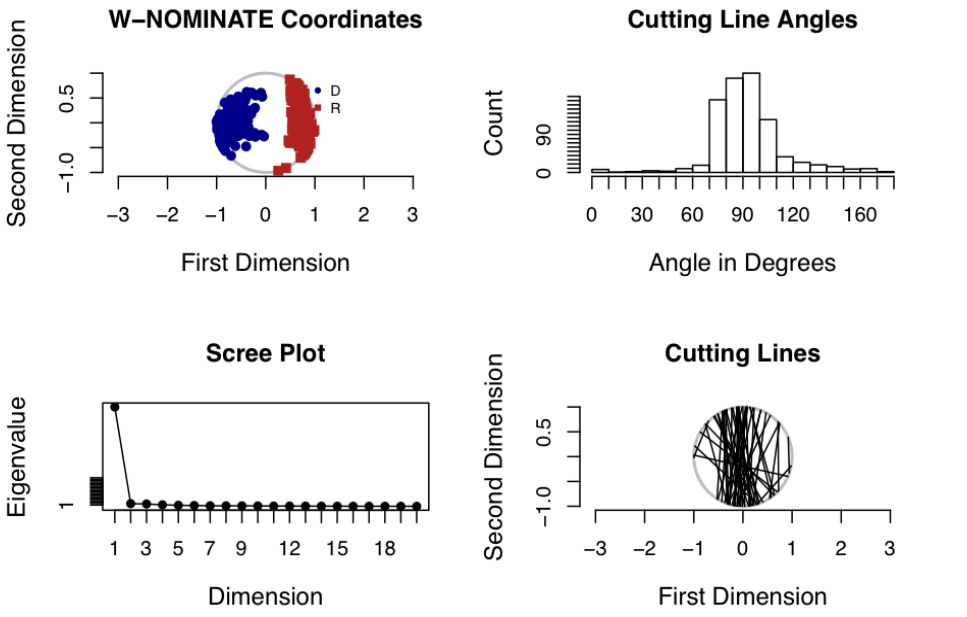


On the W-nominate plot we can see that there is a clear ideological split between republicans and democrats

(there is no overlap and a marked spatial difference between them). Within the parties themselves there is clearly more interparty ideological difference democrats than republicans, as shown by the wider spread of Ds than Rs. Further, from looking at the general density of the Ds, the average democrat seems to have (relatively) a more pro civil rights stance, although in both parties there is certainly a spread along the axis.

2. Discuss the dimensionality of the space. You can present and inspect fit via the aggregate proportion reduction in errors (APRE), the geometric mean prediction (GMP) rate, scree plots, or any other diagnostic tool (visual or numeric) to inspect the overall fit of the algorithm.

```
# canned plot(s)
plot(wnom_result)
```




```
## NULL
```

```
par(mfrow = c(1,1)) # reset plot pane space
```

From the scree plot we can see that there is one dimension that explains a signficant amount of the variation between roll call votes, and a second dimension (slightly above 1) that explains slightly more of the varition. After that, the eigenvalues go to zero and therefor lack useful explainatory power.

3. Based on the previous inspection, discuss the advantages and disadvantages of major unfolding approaches (discussed in class) to studying roll-call voting and binary choice data more generally? When might one approach perform better than other approaches? Etc.

The above example of roll call voting works well because for the spatial unfolding techniques because the complex driving forces behind voting decisions can be shown to be mapped to a very small number of dimensions (eg see the scree plot analysis above). A core part of this analysis hinges upon visual interpretability, which if you get beyond three dimensions would get uninterpretable very quickly. Obviously you could do some sort of principle components work and force things to the lower dimensions anyway, but in more complex scenarios (perhaps, hypothetically, voters preferences for political candidates are a mishmash of equally important factors involving the current state of the economy, the race/gender of the candidate, the candidates opinion on abortion, etc.) you would not able to represent these systems with these unfolding

tools.

Further, as stated in the lecture notes, for spatial scaling methods to work, we need to be able plot relative points. Points can only be coherently relative to each other under scenarios where preferences are transitive (eg if A > B, and B > C, then A must be > C). If this doesn't hold, we also cannot do unfolding approaches.

A final minor point, if we don't have the domain knowledge to be able to note extremes in the data then algorithm's like W-NOMINATE will not preform as well.